

CLAIMS

What is claimed is:

- 5 1. In a multitasking system executing real-time harmonic and dynamic tasks that can request activation or deactivation at any time, a method of scheduling tasks comprising:
- assigning priority levels to tasks;
- determining available slack for tasks at each priority level, taking into
- 10 account tasks that are activating and inactivating; and
- allocating slack to tasks in order of priority.
2. The method of claim 1, wherein tasks are scheduled according to a rate monotonic algorithm.
- 15 3. The method of claim 1, wherein an aperiodic high priority task can steal slack from available slack without impacting an execution deadline of a periodic low priority task.
- 20 4. The method of claim 1, wherein determining available slack comprises:
- determining slack consumed;
- determining timeline slack;
- determining reclaimed slack; and
- determining idle time.
- 25 5. The method of claim 4, wherein determining timeline slack comprises maintaining a table that is recalculated at task activation and deactivation.

6. The method of claim 4, wherein determining available slack comprises maintaining accumulators for slack consumed, reclaimed slack, and idle time.

7. The method of claim 6, wherein tasks have periods, and wherein maintaining
5 the accumulators comprises updating the accumulators upon the occurrence of an event from the group consisting of:

when crossing a period boundary;

when a task completes for period when executing on a fixed budget with
slack to be reclaimed;

10 when a processor executing the tasks transitions from idle to busy;
when a task completes for period when executing on slack; and
prior to calculating available slack for a new slack-consuming task.

8. The method of claim 6, wherein determining available slack comprises
15 predecrementing accumulators to allow for overhead associated with allocating slack.

9. The method of claim 1, wherein the multitasking system is a flight control
system.

20 10. A machine-readable medium having instructions stored thereon capable of causing a processor to carry out a method, the method comprising:

assigning priority levels to tasks;

determining available slack for tasks at each priority level, taking into

25 account tasks that are activating and inactivating; and

allocating slack to tasks in order of priority.

11. In a multitasking system executing real-time harmonic and dynamic tasks which can request activation or deactivation at any time, a method of scheduling tasks comprising:

- 5 determining available slack for tasks, taking into account tasks that are
activating and inactivating; and
allocating slack to requesting tasks.

12. The method of claim 11, wherein tasks are scheduled according to a rate monotonic algorithm.

10

13. The method of claim 11, wherein each task has an assigned priority, and wherein an aperiodic high priority task can steal slack from available slack without impacting an execution deadline of a periodic low priority task.

- 15 14. The method of claim 11, wherein determining available slack comprises:
determining slack consumed;
determining timeline slack;
determining reclaimed slack; and
determining idle time.

20

15. The method of claim 14, wherein determining timeline slack comprises maintaining a table that is recalculated at task activation and deactivation.

16. The method of claim 14, wherein determining available slack comprises
25 maintaining accumulators for slack consumed, reclaimed slack, and idle time.

17. The method of claim 16, wherein tasks have periods, and wherein maintaining the accumulators comprises updating the accumulators upon the occurrence of an event from the group consisting of:

when crossing a period boundary;

5 when a task completes for period when executing on a fixed budget with slack to be reclaimed;

when a processor executing the tasks transitions from idle to busy;

when a task completes for period when executing on slack; and

prior to calculating available slack for a new slack-consuming task.

10

18. The method of claim 16, wherein determining available slack comprises predecrementing accumulators to allow for overhead associated with allocating slack.

15 19. The method of claim 11, wherein the multitasking system is a flight control system.

20. A machine-readable medium having instructions stored thereon capable of causing a processor to carry out a method, the method comprising:

20 determining available slack for tasks, taking into account tasks that are activating and inactivating, by:

determining slack consumed;

determining timeline slack;

determining slack reclaimed; and

25 determining idle time;

and

allocating slack to requesting tasks.

21. A multitasking system, comprising:
a processor;
a plurality of tasks operating on the processor, wherein each task of the plurality of tasks is of a task type selected from the group consisting of periodic and
5 aperiodic, wherein each task of the plurality of tasks has associated with it at least one worst-case execution time; and
an executive in communication with the processor and controlling dispatching of tasks on the processor, wherein the executive comprises:
a first module that determines available slack, taking into account aperiodic
10 tasks that are requesting activation and deactivation at unpredictable times; and
a second module that allocates available slack to aperiodic tasks.
22. The multitasking system of claim 21, wherein the first module determines available slack by determining slack consumed, timeline slack, reclaimed slack, and
15 idle time.
23. The multitasking system of claim 21, wherein the executive controls the dispatching of tasks according to a rate monotonic algorithm.
- 20 24. The multitasking system of claim 21, wherein the first module determines timeline slack by maintaining a table and recalculating same when aperiodic tasks are activated and deactivated.
25. The multitasking system of claim 21, and further comprising accumulators to
25 store values representing slack consumed, reclaimed slack, and idle time, respectively.

26. The multitasking system of claim 25, wherein tasks have periods, and wherein the processor updates accumulators upon the occurrence of an event from the group consisting of:

when crossing a period boundary;

5 when a task completes for period when executing on a fixed budget with slack to be reclaimed;

when the processor transitions from idle to busy;

when a task completes for period when executing on slack; and

prior to calculating available slack for a new slack-consuming task.

10

27. The multitasking system of claim 25, wherein the first module predecrements accumulators to allow for overhead.

28. The multitasking system of claim 21, wherein the multitasking system is a
15 flight control system.

29. In a multitasking system executing real-time processes that can request activation or deactivation at any time, a method of scheduling processes comprising:

assigning priority levels to processes;

20 determining available slack for processes at each priority level, taking into account processes that are activating and inactivating; and

allocating slack to processes in order of priority.

30. The method of claim 29, wherein processes are scheduled according to a rate
25 monotonic algorithm.

31. The method of claim 29, wherein an aperiodic high priority process can steal slack from available slack without impacting an execution deadline of a periodic low priority process.

5 32. The method of claim 29, wherein determining available slack comprises:
determining slack consumed;
determining timeline slack;
determining reclaimed slack; and
determining idle time.

10

33. The method of claim 32, wherein determining timeline slack comprises maintaining a table that is recalculated at process activation and deactivation.

15 34. The method of claim 32, wherein determining available slack comprises maintaining accumulators for slack consumed, reclaimed slack, and idle time.

35. The method of claim 34, wherein processes have periods, and wherein maintaining the accumulators comprises updating the accumulators upon the occurrence of an event from the group consisting of:

20 when crossing a period boundary;
when a process completes for period when executing on a fixed budget with slack to be reclaimed;
when a processor executing the processes transitions from idle to busy;
when a process completes for period when executing on slack; and
25 prior to calculating available slack for a new slack-consuming process.

